J017 Rec'd PCT/PTO 25 APR 2001

FORM PTO-1390 DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE (REV 11-2000)		ATTORNEY'S DOCKET NO. 851663.424USPC
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371		U.S. APPLICATION NO (If known, see37 CFR 1.5) Unknown 09/830441
INTERNATIONAL APPLICATION NO. **PCT/SG98/00084**	INTERNATIONAL FILING DATE **26 October 1998 (24.10.1998)**	PRIORITY DATE CLAIMED **26 October 1998 (26.10.1998)**

TITLE OF INVENTION
MULTI-PRECISION TECHNIQUE FOR DIGITAL AUDIO ENCODER

APPLICANT(S) FOR DO/EO/US
ABSAR, Mohammed; GEORGE, Sapna; and ALVAREZ-TINOCO, Antonio, Mario

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (21) indicated below.
4. ☒ The US has been elected by the expiration of 19 months from the priority date (Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2)).
   a. ☒ is attached hereto (required only if not communicated by the International Bureau).
   b. ☐ has been communicated by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☐ An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
   a. ☐ is attached hereto
   b. ☐ has been previously submitted under 35 U.S.C. 154(d)(4).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3)).
   a. ☐ are attached hereto (required only if not communicated by the International Bureau).
   b. ☐ have been communicated by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. ☒ A English language translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11 to 20 below concern document(s) or information included:

11. ☐ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☐ A FIRST preliminary amendment.
14. ☐ A SECOND or SUBSEQUENT preliminary amendment.
15. ☐ A substitute specification.
16. ☐ A change of power of attorney and/or address letter.
17. ☐ A computer-readable form of the sequence listing in accordance with PCT Rule 13ter.2 and 35 U.S.C. 1.821 – 1.825.
18. ☒ A second copy of the published international application under 35 U.S.C. 154(d)(4)
19. ☐ A second copy of the English language translation of the international application under 35 U.S.C. 154(d)(4).
20. ☐ Other items of information:

531 Rec'd PCT/PTO 25 APR 2001

U.S. APPLICATION NO. (If known, see 37 CFR 1.5) **Unknown** 09/830441	INTERNATIONAL APPLICATION NO. **PCT/SG98/00084**	ATTORNEY'S DOCKET NUMBER **851663.424USPC**

21. ☒ The following fees are submitted:

Basic National Fee (37 CFR 1.492(a)(1)-(5)):

CALCULATIONS PTO USE ONLY

Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO $1000.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO $860.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO $710.00

International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(1)-(4)..................... $690.00

International preliminary examination fee (37 CFR 1.482) paid to USPTO and all claims satisfied provisions of PCT Article 33(1)-(4) $100.00

ENTER APPROPRIATE BASIC FEE AMOUNT =				$860.00	
Surcharge of $130.00 for furnishing the oath or declaration later than ☐ 20 ☒ 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				$130.00	
Claims	Number Filed	Number Extra	Rate		
Total Claims	38 - 20 =	18	x $ 18.00	$324.00	
Independent Claims	2 - 3 =	0	x $ 80.00	$.00	
Multiple dependent claim(s) (if applicable)			+ $270.00	$270.00	
TOTAL OF ABOVE CALCULATIONS =				$1584.00	
☐ Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above are reduced by 1/2.				$.00	
SUBTOTAL =				$1584.00	
Processing fee of $130.00 for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492(f)). +				$.00	
TOTAL NATIONAL FEE =				$1584.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). $40.00 per property +				$.00	
TOTAL FEES ENCLOSED =				**$1584.00**	
				Amount to be refunded:	
				charged	

a. ☒ A check in the amount of $1584.00 cover the above fees is enclosed.

b. ☐ Please charge my Deposit Account No. in the amount of $_ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. **19-1090**. A duplicate copy of this sheet is enclosed.

d. ☐ Fees are to be charged to a credit card. WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

GASH, Eric, J.
Seed Intellectual Property Law Group PLLC
701 5th Avenue, Suite 6300
Seattle, WA 98104-7092
United States of America
(206) 622-4900

Eric Gash
SIGNATURE

Eric J. Gash
NAME

46,274
REGISTRATION NUMBER

7 PRTS

09/830441
531 Rec'd PCT/PTO 25 APR 2001

MULTI-PRECISION TECHNIQUE FOR DIGITAL AUDIO ENCODER

Field of the Invention

This invention is applicable in the field of audio encoders, and in particular to those audio encoders which may be implemented on fixed point arithmetic digital processors, such as for professional and commercial applications.

Background of the Invention

In order to more efficiently broadcast or record audio signals, the amount of information required to represent the audio signals may be reduced. In the case of digital audio signals, the amount of digital information needed to accurately reproduce the original pulse code modulation (PCM) samples may be reduced by applying a digital compression algorithm, resulting in a digitally compressed representation of the original signal. The goal of the digital compression algorithm is to produce a digital representation of an audio signal which, when decoded and reproduced, sounds the same as the original signal, while using a minimum of digital information for the compressed or encoded representation.

Recent advances in audio coding technology have led to high compression ratios while keeping audible degradation in the compressed signal to a minimum. These coders are intended for a variety of applications, including 5.1 channel film soundtracks, HDTV, laser discs and multimedia. Description of one applicable method can be found in the Advanced Television Systems Committee (ATSC) Standard document entitled "Digital Audio Compression (AC-3) Standard", Document A/52, 20 December, 1995, and the disclosure of that document is hereby expressly incorporated herein by reference.

The implementation of an AC-3 encoder by translation of the requirements and processes from the abovementioned AC-3 Standard onto the firmware of a Digital Signal Processor (DSP) core involves several phases. Firstly, the essential compression algorithm blocks of the AC-3 Encoder have to be designed, since it is only the functions which are defined by the standard. After individual blocks are completed, they are integrated into an encoding system

which receives a PCM (pulse code modulated) stream, processes the signal applying signal processing techniques such as transient detection, frequency transformation, masking and psychoacoustic analysis, and produces a compressed stream in the format of the AC-3 Standard.

The coded AC-3 stream should be capable of being decompressed by any standard AC-3 Decoder and the PCM stream generated thereby should be comparable in audio quality to the original music stream. If the original stream and the decompressed stream are indistinguishable in audible quality (at reasonable level of compression) the development moves to the third phase. If the quality is not transparent (indistinguishable), further algorithmic development and improvements continue.

In the third phase the algorithms are simulated in a high level language (e.g. C) using the word-length specifications of the target DSP-Core. Most commercial DSP-Cores allow only fixed point arithmetic (since a floating point engine is costly in terms of integrated circuit area). Consequently, the encoder algorithms are translated to a fixed point solution. The word-length used is usually dictated by the ALU (arithmetic-logic unit) capabilities and bus-width of the target core. For example, an AC-3 encoder on a Motorola 56000 DSP would use 24-bit precision since it is a 24-bit Core. Similarly, for implementation on a Zoran ZR38000 which has a 20-bit data path, 20-bit precision would be used.

If, for example, 20-bit precision is discovered to provide an unacceptable level of sound quality, the provision to use double precision always exists. In this case each piece of data is stored and processed as two segments, lower and upper words, each of 20-bit length. The accuracy of implementation is doubled but so is the computational complexity, and double precision multiplication could require 6 or more cycles where a single precision multiplication and addition *(MAC)* may use only a single cycle. Block exponent and other boosting techniques specific to the AC-3 encoder can be judiciously used to improve the quality, but these features are not always found on the commercial DSPs.

Single precision 24-bit AC-3 encoders are known to provide sufficient quality. However 16-bit single precision AC-3 encoder quality is considered very poor. Consequently, the implementation of AC-3 encoders on 16-bit DSP cores has not been popular. Since a single precision 16-bit implementation of an AC-3 encoder results in unacceptable in reproduction quality, such a product would be at a distinct disadvantage in the consumer market. On the other hand, double precision implementation is too computationally expensive. It has been estimated that a fully double precision implementation would require over 140 MIPS (million instruction per second). This exceeds what most commercial DSPs can provide, and moreover, extra MIPS are always needed for system software and value-added features.

Summary of the Invention

In accordance with the present invention, there is provided a method for coding digital audio data with a transform encoding process implemented on a fixed point digital signal processor having multiple levels of computation precision, wherein the transform encoding process includes a plurality of computation stages involving arithmetic operations in transforming the digital audio data into coded audio data, and wherein different ones of the computation stages utilise different preselected levels of computational precision, characterised in that:

the transform encoding process is in accordance with AC-3 Digital Audio Compression Standard.

The present invention also provides a digital audio transform encoder for coding digital audio data into compressed audio data, comprising a fixed point digital signal processor having multiple levels of computation precision, and transform encoding process code stored in firmware or software for controlling the digital signal processor, wherein the transform encoding process code includes a plurality of computation blocks involving arithmetic operations in transforming the digital audio data into compressed audio data, and wherein different ones of the computation blocks are performed by the digital signal processor using different preselected levels of computational precision, characterised in that:

the transform encoding process code is in accordance with AC-3 Digital Audio Compression Standard.

In a preferred form of the invention, the audio transform encoding system is implemented on a 16-bit digital signal processor which is capable of single (16-bit) precision computations and double (32-bit) computations. Accordingly, the preferred 16-bit implementation uses

combinations of single and double precision to best match the reference floating point model. Thereby, computational complexity is reduced without sacrificing quality excessively. The features of the preferred embodiment which are discussed in general terms below are thus presented in the context of such an implementation.

For transient detection, single precision (16-bit) calculations can be used. The input stream of PCM audio data is assumed to be 16 bits (else it is truncated to 16 bits for this stage) and the high pass filter coefficients are restricted to 16-bits as well. The filtered 16-bit data is segmented and analysed to detect transients. Simulation results with music streams indicate that the result of this implementation matches over 99% of the time with the floating point version. Since this step involves only 16-bit operations it is termed as 16-16 (data:coefficient) processing.

The input 16-bit PCM is transformed to the frequency domain by first applying a window with 32-bit length coefficients. Therefore windowing is 16-32 (data:coefficient) processing. If the input PCM is 24-bit, then 32-16 processing for windowing may be used wherein the PCM data is treated as 32-bit (upper bits sign extended) and is multiplied by 16-bit window coefficients.

Frequency Transformation using Modified Discrete Cosine Transform (MDCT) is performed using 32-bit data and 16-bit coefficients. For each calculation, the input data is 32-bit and is multiplied by the coefficients (sine and cosine terms) which are 16-bit in length. The resulting 48-bit is truncated to 32-bit for the next step of processing. This form of frequency transformation with 32-16 processing can be shown to give 21-25 bit accuracy with 80% confidence, when compared with the floating point version.

Each 32-bit frequency coefficient is assumed to be stored in two 16-bit registers. For phase and coupling strategy calculations the upper 16-bit of the data can be utilised. Once the strategy for combining the coupled channel to form the coupling channel is known, the combining process uses the full 32-bit data. The computation is reduced while the accuracy

is still high. Simple truncation of the upper 16-bit of the 32-bit data for the phase and coupling strategy calculation leads to poor result (only 80% of the time the strategy matches with that from the floating point version), and thus a block exponent pre-processing method can be employed. If the block exponent method is used the coupling strategy is 97% of the time exactly same as the floating point.

A rematrixing decision determines whether to code coefficients as left (L) and right channel (R), or the sum (L+R) and difference (L - R) of the channels, and can be made using the upper 16-bit of the 32-bit data. The actual rematrix coding of coefficients preferably uses the full 32-bit data as in the coupling calculations.

The remaining processing of the AC-3 encoding, including exponent coding, quantization and bit allocation are defined as fixed point arithmetic in the AC-3 Standard and therefore word-length choices are not encountered in these calculations.

Brief Description of the Drawings

The invention is described in greater detail hereinafter, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a system block diagram of an AC-3 compliant audio encoder;

Figure 2 is a comparison of 24-24 (data:coefficient) bit and 16-16 (data:coefficient) bit wordlengths with floating point calculations for transient detection;

Figure 3 is a flow diagram of a transient detection process, wherein 16-32 (data-coefficient) bit precision is used for windowing operations while 32-16 (data-coefficient) is used for frequency transformation;

Figure 4 shows comparative charts of error probability of fixed point (32-16 & 24-24), with the floating-point calculation as reference, for the frequency transformation stage;

Figure 5 is a block diagram illustrating coupling coefficient generation and phase estimation;

Figure 6 is a diagram illustrating block exponent processing;

Figures 7, 8 and 9 are frequency response charts of AC-3 encoder implementation in terms

of signal-to-noise ratio for floating point, 16-32 bit and 24 bit calculations, respectively.

Detailed Description of the Preferred Embodiments

In the following detailed description of the preferred embodiments of the invention, firstly a system-level description of an AC-3 encoder is provided. This serves to explain the overall processes and describe the significance of each processing block in the overall audio compression system.

After the system level description, the word-length requirements of each processing blocks, where fixed point arithmetic is used, is discussed. This includes the transient-detection, frequency transformation, rematrixing and coupling blocks. By analysis of data gathered through extensive simulation, and statistics derived thence, appropriate word-length requirements for each block are then estimated. In particular, this description deals with the issue of the implementation of the dual-channel AC-3 encoder on a 16-bit processor in a manner such that the processing requirement is not prohibitive and the quality is comparable to implementation on single precision 24-bit processor.

System Overview

Like the AC-2 single channel coding technology from which it is derived, an AC-3 audio coder is fundamentally an adaptive transform-based coder using a frequency-linear, critically sampled filter bank based on the Princen Bradley Time Domain Aliasing Cancellation (TDAC) technique. An overall system block diagram of an AC-3 coder 10 is shown in Figure 1. It may be noted that, of the blocks shown in Figure 1, blocks such as the Frame Optimisation Tables 22, Fast Bit Allocation 21 and Spectral Reshaping 18 are not directly part of the AC-3 Standard but are desirable for high quality audio reproduction and for reducing the computational burden.

Audio Input Format

AC-3 is a block structured coder, so one or more blocks of time domain signal, typically *512* samples per block and channel, are collected in an input buffer before proceeding with

additional processing.

Transient Detection

Transients are detected in the full-bandwidth channels in order to decide when to switch to short length audio blocks for restricting quantization noise associated with the transient within a small temporal region about the transient. The input audio signals are high-pass filtered (12), and then examined by a transient detector (13) for an increase in energy from one sub-block time segment to the next. Sub-blocks are examined at different time scales. If a transient is detected in the second half of an audio block in a channel, that channel switches to a short block (256 samples). In presence of a transient the bit *'blksw'* for the channel in the encoded bit stream in the particular audio block is set.

The transient detector operates on 512 samples for every audio block. This is done in two passes, with each pass processing 256 samples. Transient detection is broken down into four steps:

1. high pass filtering;
2. segmentation of the block into sub-multiples;
3. peak amplitude detection within each sub-block segment; and
4. threshold comparison.

The transient detector outputs the flag *blksw* for each full-bandwidth channel, which when set to 'one' indicates the presence of a transient in the second half of the *512* length input block for the corresponding channel. The four stages of the transient detection are described in further detail below.

1) High pass filtering: The high-pass filter can be implemented as a cascade biquad direct form II IIR filter with a cut-off of 8 kHz.

2) Block Segmentation. The block of *256* high-pass filtered samples are segmented into a hierarchical tree of levels in which level 1 represents the *256* length block, level 2 is two

segments of length *128*, and level 3 is four segments of length *64*.

3) Peak Detection: The sample with the largest magnitude is identified for each segment on every level of hierarchical tree. The peaks for a single level are found as follows:

$$P[i][j] = max(x(n))$$

$$\text{for } n = (512 \times (k-l)/2\hat{\ }j) + 1, \ldots\ldots (512 \times k/2\hat{\ }j) - 1$$

$$\text{and } k = 1, \ldots\ldots, 2\hat{\ }(j-l);$$

where $x(n)$ = *the nth sample in the 256 length block*

$j = 1, 2, 3$ *is the hierarchical level number*

k = *the segment number within level j*

4) Threshold comparison: The first stage of the threshold comparator checks to see if there is significant signal level in the current block. This is done by comparing the overall peak value *P[1][1]* of the current block to a "silence threshold". If *P[1][1]* is below this threshold then a long block is forced. The silence threshold value is *100/32768*. The next stage of the comparator checks the relative peak levels of adjacent segments on each level of the hierarchical tree. If the peak ratio of any two adjacent segments on a particular level exceeds a pre-defined threshold for that level, then a flag is set to indicate the presence of a transient in the current 256 length block.

Time Domain Aliasing Cancellation (TDAC) Filter Bank

The time domain input signal for each channels is individually windowed and filtered with a TDAC-based analysis filter bank (11) to generate frequency domain coefficients. If the *blksw* bit is set, meaning that a transient was detected for the block, then two short transforms of length *256* each are taken, which increases the temporal resolution of the signal. If *blksw* is not set, a single long transform of length *512* is taken, thereby providing a high spectral resolution.

The output frequency sequence [k] is defined as :

$$X_k = \sum_{n=N-1}^{n=0} x[n] * \cos(2\pi * (2n+1) * (2k+1)/4N + \pi * (2k+1)/4) \qquad k=0..(N/2-1)$$

where $x[n]$ is the windowed input sequence for a channel and N is the transform length.

Instead of evaluating X_k in the form given above it can be computed in a computationally efficient manner as described in the specification of International Patent Application No. PCT/SG98/00014 entitled "*A Fast Frequency Transformation Technique for Transform Audio Coders*" The disclosure of that document is hereby expressly incorporated herein by reference, and an explanatory extract is presented below:

Instead of evaluating X_k in the form given above it could be computed as

$$\begin{aligned} X_k = {} & \cos\gamma * (g_{k,r}\cos(\pi(k+1/2)/N) - g_{k,i}\sin(\pi(k+1/2)/N)) \\ & -\sin\gamma * (g_{k,r}\sin(\pi(k+1/2)/N) + g_{k,i}\cos(\pi(k+1/2)/N)) \\ & g_{k,r}, g_{k,i} \in \Re(\textit{set of real numbers}) \end{aligned}$$

where $G_k = g_{k,r} + jg_{k,i} = \sum_{n=0}^{n=N-1} (x[n]e^{j\pi n/N}) * e^{j2\pi nk/N}$. The symbol j represents the imaginary number $\sqrt{-1}$. The expression $\sum_{n=0}^{n=N-1} (x[n]e^{j\pi n/N}) * e^{j2\pi nk/N}$ is obtained from the well known FFT method, by first using transformation $x'[n] = x[n] * e^{j\pi n/N}$ and then computing the FFT $G_k = \sum_{n=0}^{n=N-1} x'[n] * e^{j2\pi nk/N}$

Coupling

High compression can be achieved in AC-3 by use of a technique known as coupling. Coupling takes advantage of the way the human ear determines directionality for high frequency signals. At high audio frequency (approximately above 4KHz), the ear is physically unable to detect individual cycles of an audio waveform and instead responds to

the envelope of the waveform. Consequently, the encoder 10 may include a coupling processor (14) which combines the high frequency coefficients of the individual channels to form a common coupling channel. The original channels combined to form the coupling channel are called the coupled channels.

The most basic encoder can form the coupling channel by simply taking the average of all the individual channel coefficients. A more sophisticated encoder could alter the signs of the individual channels before adding them into the sum to avoid phase cancellation.

The generated coupling channel is sectioned into a number of bands. For each such band and each coupling channel a coupling co-ordinate is transmitted to the decoder. To obtain the high frequency coefficients in any band, for a particular coupled channel, from the coupling channel, the decoder multiplies the coupling channel coefficients in that frequency band by the coupling co-ordinate of that channel for that particular frequency band. For a dual channel encoder a phase correction information is also sent for each frequency band of the coupling channel.

Superior methods of coupling channel formation are discussed in the specification of International Patent Applications PCT/SG97/00076, entitled "*Method and Apparatus for Estimation of Coupling Parameters in a Transform Coder for High Quality Audio*", and PCT/SG97/00075 entitled "*Method and Apparatus for Phase Estimation in a Transform Coder for High Quality Audio*". The disclosures of those specifications are hereby expressly incorporated herein by reference. An explanatory extract from the latter specification is presented below for reference.

"Assume that the frequency domain coefficients are identified as:

a_i , for the first coupled channel ,

b_i , for the second coupled channel ,

c_i , for the coupling channel ,

For each sub-band, the value $\sum_i a_i * b_i$ is computed, index i extending over the frequency

range of the sub-band. If $\sum_i a_i * b_i > 0$, coupling for this sub-band is performed as $c_i = (a_i + b_i)/2$. Similarly, if $\sum_i a_i * b_i < 0$, then coupling strategy for the sub-band is performed as $c_i = (a_i - b_i)/2$.

Adjacent sub-bands using identical coupling strategies may be grouped together to form one or more coupling bands. However, sub-bands with different coupling strategies must not be banded together. If overall coupling strategy for a band is $c_i = (a_i + b_i)/2$, i.e. for all sub-bands comprising the band the phase flag for the band is set to +1, else it is set to -1."

Rematrixing

An additional process, rematrixing (15), is invoked in the special case that the encoder is processing two channels only. The sum and difference of the two signals from each channel are calculated on a band by band basis, and if, in a given band, the level disparity between the derived (matrixed) signal pair is greater than the corresponding level of the original signal, the matrix pair is chosen instead. More bits are provided in the bit stream to indicate this condition, in response to which the decoder performs a complementary unmatrixing operation to restore the original signals. The rematrix bits are omitted if the coded channels are more than two.

The benefit of this technique is that it avoids directional unmasking if the decoded signals are subsequently processed by a matrix surround processor, such a Dolby Prologic (TM) decoder.

In AC-3, rematrixing is performed independently in separate frequency bands. There are four bands with boundary locations dependent on the coupling information. The boundary locations are by coefficient bin number, and the corresponding rematrixing band frequency boundaries change with sampling frequency.

Conversion to Floating Point

The transformed values, which may have undergone rematrix and coupling process, are

converted to a specific floating point representation at the exponent extraction block (16), resulting in separate arrays of binary exponents and mantissas. This floating point arrangement is maintained through out the remainder of the coding process, until just prior to the decoder's inverse transform, and provides 144 dB dynamic range, as well as allows AC-3 to be implemented on either fixed or floating point hardware.

Coded audio information consists essentially of separate representation of the exponent and mantissa arrays. The remaining coding process focuses individually on reducing the exponent and mantissa data rate.

The exponents are coded using one of the exponent coding strategies. Each mantissa is truncated to a fixed number of binary places. The number of bits to be used for coding each mantissa is to be obtained from a *bit allocation algorithm* which is based on the masking property of the human auditory system.

Exponent Coding Strategy

Exponent values in AC-3 are allowed to range from *0* to *-24*. The exponent acts as a scale factor for each mantissa, equal to 2^{-exp}. Exponents for coefficients which have more than *24* leading zeros are fixed at *-24* and the corresponding mantissas are allowed to have leading zeros.

AC-3 bit stream contains exponents for independent, coupled and the coupling channels. Exponent information may be shared across blocks within a frame, so blocks 1 through 5 may reuse exponents from previous blocks.

AC-3 exponent transmission employs differential coding technique, in which the exponents for a channel are differentially coded across frequency. The first exponent is always sent as an absolute value The value indicates the number of leading zeros of the first transform coefficient. Successive exponents are sent as differential values which must be added to the prior exponent value to form the next actual exponent value.

The differential encoded exponents are next combined into groups. The grouping is done by one of the three methods: *D15*, *D25* and *D45*. These together with *'reuse'* are referred to as exponent strategies. The number of exponents in each group depends only on the exponent strategy. In the *D15* mode, each group is formed from three exponents. In *D45* four exponents are represented by one differential value. Next, three consecutive such representative differential values are grouped together to form one group. Each group always comprises of 7 bits. In case the strategy is 'reuse' for a channel in a block, then no exponents are sent for that channel and the decoder reuses the exponents last sent for this channel.

Pre-processing of exponents prior to coding can lead to better audio quality. One such processing technique is described in the specification of International Patent Application PCT/SG98/0002 entitled *"Method and Apparatus for Spectral Exponent Reshaping in a Transform Coder for High Quality Audio"*, the disclosure of which is incorporated herein by reference.

Choice of the suitable strategy for exponent coding forms an important aspect of AC-3, and in the encoder 10 shown in Figure 1 is performed by the process blocks 17, 18. *D15* provides the highest accuracy but is low in compression. On the other hand transmitting only one exponent set for a channel in the frame (in the first audio block of the frame) and attempting to *'reuse'* the same exponents for the next five audio block, can lead to high exponent compression but also sometimes very audible distortion.

Several methods exist for determination of exponent strategy, and one such method is described in the specification of International Patent Application no. PCT/SG98/00009 entitled *"A Neural Network Based Method for Exponent Coding in a Transform Coder for High Quality Audio"*.

Bit Allocation for Mantissas

The bit allocation algorithm (block 21) analyses the spectral envelope of the audio signal being coded, with respect to masking effects, to determine the number of bits to assign to

each transform coefficient mantissa. In the encoder, the bit allocation is recommended to be performed globally on the ensemble of channels as an entity, from a common bit pool.

The bit allocation routine contains a parametric model (psycho-acoustic analysis block 20) of the human hearing for estimating a noise level threshold, expressed as a function of frequency, which separates audible from inaudible spectral components. Various parameters of the hearing model can be adjusted by the encoder depending upon the signal characteristics. For example, a prototype masking curve is defined in terms of two piecewise continuous line segments, each with its own slope and y-intercept.

Word-Length Requirements of Processing Blocks

Floating point arithmetic usually uses the procedures set out in IEEE 754 (i.e. 32 bit representation, with 24-bit mantissa, 7-bit exponent & 1 sign bit) which is adequate for high quality AC-3 encoding Work-stations like *Sun SPARCstation 20* (TM) can provide much higher precision (e.g. double precision is 8 bytes). However floating point units require greater integrated circuit area and consequently most DSP Processors use fixed point arithmetic. The AC-3 encoder, in use, is often intended to be a part of a consumer product e.g. DVDRAM (Digital Versatile Disk Readable and Writeable) where cost (chip area) is an important factor.

Being aware of the cost versus quality issue in the development of the AC-3 standard, Dolby Laboratories has ensured that the algorithms can be implemented on fixed-point processors, however an important issue is what word-length is required of the fixed-point processor for processing high quality audio signals.

The AC-3 encoder has been implemented on 24-bit processors such as the Motorola 56000 and has met with much commercial success. However, although the performance of an AC-3 encoder implemented on a 16-bit processor is universally assumed to be of low quality, no adequate study has been conducted to benchmark the quality or compare it with the floating point version.

As discussed above, using double precision (32-bit) to implement the encoder on a 16-bit processor can lead to high quality (even more than a 24-bit processor implementation). However, double precision arithmetic is very computationally expensive (e.g. on D950 single precision multiplication takes 1 cycle whereas double precision requires 6 cycles). Accordingly, rather than performing single or double precision throughout the whole encoding process, an analysis can be performed to determine adequate precision requirements for each stage of computation.

In the description that follows, for simplicity of expression (and to avoid repetition), the following convention has been adopted. Notation *x-y (set A:set B)* implies that for the process, data elements within *Set A* are limited or truncated to *x* bits while the *Set B* elements are *y* bits long. For example, *16-32* (data:window) implies that, for windowing, data was truncated to *16* bits and the window coefficient to 32 bits. When appearing without any parenthesised explanation, e.g. *x-y* : explanation of the implied meaning is generally provided. If no explanation is provided the meaning will be clear from the context, and the brevity of expression has taken precedence over repetition of the same idea.

Based on extensive simulations and study of the statistics derived thereon, it has been determined that the different stages of the encoder can be suitably implemented with different combinations of computational precision, such as : 16-32, 32-16, 16-16 and 32-32. Suitable trade-off in terms of MIPS and quality are therefore made subject to the statistics obtained, and the computational strategies which may be adopted for various processing stages as a result are discussed below.

Transient Detection

In a simulation, the high-pass filtering and the subsequent segment analysis for transient detection was performed with 16 and 24-bit word-lengths (both single precision). The input PCM stream is assumed to be 16-bit and the filter coefficients are truncated to 16 bits also The output of the filter is a 16-bit number which is analysed for transients. Thus, this process is entirely 16-16 (data:coefficient).

The simulation results are compared with the floating point version resulting from processing with a *Sun SparcStation 20* (TM). For the simulation, five music samples were used, namely: *a) Drums, b) Harp, c) Piano, d) Saxophone and e) Vocal*. Although not exhaustive, it is believed that these are sufficient to provide a good example of complex audio streams. Figure 2 of the accompanying drawings is a graph of transient detection, with a comparison of 16-16 (data:coefficient) and 24-24 (data:coefficient) wordlengths with the floating point results. As is evident from the chart, the 16-16 result matches over 99% of the time with the floating point.

From Figure 2 it is evident that for more than 99% cases, the 16-bit output from Transient Detection 13 (in terms of the *blksw* information) is same as the floating point version. This implies that for this stage of processing double precision computation adds little benefit, and 16-bit single precision is adequate. Simulation results for 24-24 (data:coefficients) are also shown int the Figure.

Forward Transform

Windowing

The audio block is multiplied by a window function to reduce transform boundary effects and to improve frequency selectivity in the filter bank 11. The values of the window function are included in ATSC specification Document referred to above. If the input audio is considered to be 16-bit then for the windowing operation the data wordlength of more than 16 is unnecessary. For implementation on a 16-bit processor the window coefficients can be 16 or 32-bit. In general, 16-bit coefficients are inadequate and it is recommended that 32-bits be used for the windowing coefficients. Moreover, this step forms the baseline for further processing and limiting accuracy at this stage is not reasonable. However, if the input stream is 24-bit then 32-16 (data:coefficient) processing can be performed.

Time to Frequency Transformation

Based on the block switch flags, each audio block is transformed into the frequency domain by performing one long *512*-point transform, or two short *256*-point transforms. Each

windowed data is 32-bit long. For the frequency transformation stage, coefficient (cosine and sine terms) length is restricted to 16-bit. Thus using previous terminology this is *32-16 (data:coefficient)* computation.

The advantage of 32-16 precision is that the computation burden is not as much as the 32-32 (pure double-precision) version. On the D950 32-16 multiplication takes 3 cycles while 32-32 requires 6 cycles.

Figure 3 illustrates a transient detection procedure which is entirely 16-bit: 16-32 (data-coefficient) bit precision is used for the windowing operation while 32-16 (data-coefficient) is used for the frequency transformation. From Figure 3, note that the windowing coefficients are *32*-bit while the input data (CD Quality) is *16*-bit. The *32*-bit window is multiplied by the *16*-bit data to generate *32*-bit data. This *32*-bit windowed signal is converted to the frequency domain using the Modified Discrete Cosine Transform (MDCT). The 32-16 precision is compared with the floating point version and the 24-24 bit version in Table 1, below, and the mean of the error and the standard deviation is tabulated.

	Drums		*Harp*		*Piano*		*Saxophone*		*Vocal*	
	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ
16-32	*0.1*	*499*	*0.03*	*122*	*0.04*	*106*	*0.02*	*104*	*0.02*	*94.3*
24-24	*0.1*	*127*	*0.13*	*128*	*0.12*	*129*	*0.1*	*127*	*0.15*	*124*

* *all data has been pre-scaled (multiplied) by* 10^{-5}.

Table 1. Frequency Transformation Stage : Mean ($\bar{e}$) and Standard Deviation (σ) of the error between floating-point and the fixed-point (32-16 & 24-24) implementations.

It can be observed that the mean error is about *0.0000005*, wherein the discrepancy is usually at the 20 binary place. However, since the standard deviation (σ) is much larger than the mean ($\bar{e}$), it reflects more truly the behaviour of the different implementations. Given the set

of observations, it is often convenient to condense and summarise the data by fitting it to a model that depends on adjustable parameters (in this case the error depends on the adjustable word-length). Therefore it is instructive to analyse the probability distribution of the error function.

Figure 4 shows two charts of error probability for the frequency transformation stage for 32-16 and 24-24 fixed point computations with the floating-point version as reference. The probability distribution is based on simulation results with sample space of 40,000. From the Figure it can be observed that 80% of the time 21 to 25 bit accuracy exists for the 32-16 implementation. For the 24-24, the same is true for the range 18 to 21 bits. Assuming Gaussian distribution for the error-function (which is reasonable, looking at the probability distribution in the figure above), it can be stated that for 32-16, 99.7% of the time the error is less than 0.005 (3σ). The low value is highly influenced by the statistics from the drums section of the audio input. For 24-24, with 99.7% confidence, the error is less than 0.003 (3σ). From Figure 4, it can also be noted that the spread of the error-function is less for 24-24 which implies a more stable performance as compared to 32-16. This figure of merit function, though not accurate at least serves to highlight that both the implementations have reasonably high accuracy.

Coupling Process

The computational requirements for the coupling process is quite appreciable, which makes selection of appropriate precision more difficult. The input to the coupling process is the channel coefficients each of 32-bit length. The coupling progresses in several stages. For each such stage appropriate word length must be determined.

Coupling Channel Generation Strategy

As discussed hereinabove, the coupling channel generation strategy is linked to the product $\sum a_i * b_i$, where a_i and b_i are the two coupled channel coefficients within the band in question. Although 32-32 (double precision) computation for the dot product would lead to more accurate results, it is also computationally prohibitive. An important issue, however, is that

the output of this stage only influences how the coupling channel is generated, not the accuracy of the coefficients themselves. If the error from 16-bit computation is not appreciably large, computational burden can be decreased.

Figure 5 is a block diagram of the coupling process 30. In the process shown in this Figure, 16-bit (upper half) single precision only is utilised for the coupling coefficient generation strategy and phase estimation. The actual coupling is then performed on the full 32-bit data. Coupling co-ordinates may be generated also using single precision.

As shown in the Figure, for phase estimation and coupling coefficient generation strategy (31), the upper 16-bits of the full 32-bit data from the frequency transformation stage may be used. The actual coupling coefficient generation of $c_i = (a_i \pm b_i)/2$ (33) is performed using 32-32 $(a_i{:}b_i)$ precision.

A similar approach of 16-16 $(a_i{:}b_i)$ is used for the coupling co-ordinate generation (34, 35) However, the final division involved in the co-ordinate generation must preferably be done with highest precision possible. For this it is recommended that the floating point operation be emulated, that is the exponents (equivalent to number of leading zero) and mantissa (remaining 16 bits after removal of leading zeros). The division can then be performed using the best possible method as provided by the processor to provide maximum accuracy Since coupling co-ordinates anyway need to be converted to floating point format (exponent and mantissa) for final transmission, this approach has dual benefit.

	Band 0		*Band 1*		*Band 2*		*Band 3*	
	16-16	*24-24*	*16-16*	*24-24*	*16-16*	*24-24*	*16-16*	*24-24*
Drums	*84.1*	*99.7*	*75*	*99.8*	*90*	*100*	*91*	*100*
Harp	*75.2*	*99.2*	*72.7*	*99.4*	*78.1*	*99.5*	*75.1*	*99.5*
Piano	*88.2*	*99.9*	*84*	*99.4*	*86*	*99.2*	*76*	*98.7*
Saxophone	*73.6*	*99.9*	*56*	*99.8*	*76.2*	*99.7*	*81.4*	*9.8*

	Band 0		Band 1		Band 2		Band 3	
Vocal	98.6	97.8	97.8	100	98.6	99.8	96.5	100

Table 2. Coupling Strategy : coupling strategy for each band with the 24-24 and the 16-16 approach are compared (in percentage %) with the floating point version. While 24-24 gives superior result, the 16-16 fares badly.

Table 2, above, illustrates comparative results of coupling strategies in bands for the simulation audio data, using the floating point calculations as a reference. The results for 16-16 are not as desired. Upon analysis of the reason for the low performance it can be shown that usually the coupling coefficients are low value. Thus, even though the coupling coefficient may be represented by 32-bits the higher 16-bits are normally almost all zeros. Therefore simple truncation of the upper 16 bits produce poor results. A variation of the block exponent strategy, discussed below, can be used to improve the results.

Figure 6 is a diagram illustrating block exponent processing, showing a pre-processing stage which can be implemented before truncation of the 32-bit to 16-bit for the phase estimation, coupling coefficient generation strategy and calculation of the coupling co-ordinates. In this procedure, the coefficients within the band (or sub-band depending on the level of processing) are analysed to find the minimum number of leading zeros (in actual implementation the maximum absolute rather than leading zeros are used for scaling). The entire coefficient set within the band is then shifted (equivalent to multiplication) to the left and then the remaining upper 16 bits are utilised for the processing. Note that for the phase estimation and coupling strategy the multiplication factor has no affect as long as both the left and right channels within the band have been shifted by same number of bits.

For the coupling co-ordinate generation phase, both the coupling and the coupled channels should have the same multiplication factor so that they cancel out. Alternately, if floating point emulation is used as recommended above, the coupling and coupled channels may be on different scale. The difference in scale is compensated in the exponent value of the final

coupling co-ordinate. Consider, for example, a that band has only 4 bins, 96...99:

a[96]=(0000 0000 0000 0000 1100 0000 0000 1001)
b[96]=(0000 0000 0000 0000 0000 0000 0000 0100)
c[96]=(0000 0000 0000 0000 0110 0000 0000 0110)

a[97]=(0000 0000 0000 0000 1100 0000 0000 0000)
b[97]=(0000 0000 0000 0000 0001 0000 0000 1000)
c[97]=(0000 0000 0000 0000 0110 1000 0000 0100)

a[98]=(0000 0000 0000 0000 0000 0000 0000 1000)
b[98]=(0000 0000 0000 0000 0000 0000 0000 1100)
c[98]=(0000 0000 0000 0000 0000 0000 0000 1010)

a[99]=(0000 0000 0000 0000 1100 0000 0000 1000)
b[99]=(0000 0000 0000 0001 0000 0000 0000 1100)
c[99]=(0000 0000 0000 0000 1110 0000 0000 1010)

*Note: for this example $c_i=(a_i+b_i)/2$

Considering only the upper 16-bits in this case will clearly lead to a poor result. For example coupling co-ordinate $\Psi_a=(\sum a^2/\sum b^2)$ will be zero, thereby wiping away all frequency components within the band for channel a when the coupling coefficient is multiplied by the coupling co-ordinate at the decoder to reproduce the coefficients for channel a. However, by removing the leading zeros, the new coefficients for channel a will be:

a[96]=(00 1100 0000 0000 10)
a[97]=(00 1100 0000 0000 00)
a[98]=(00 0000 0000 0000 10)
a[99]=(00 1100 0000 0000 10)

on which more meaning measurements can be performed. The scaling factor will have to be

compensated in the exponent value for the coupling co-ordinate With this approach the performance of phase estimation with 16-16 bit processing improves drastically as illustrated by the results shown in Table 3. as compared to the figures in Table 2.

	Band 0		Band 1		Band 2		Band 3	
	16-16	24-24	16-16	24-24	16-16	24-24	16-16	24-24
Drums	100	99.7	99.8	99.8	100	100	99	100
Harp	99.7	99.2	99.4	99.4	99.5	99.5	99.57	99.5
Piano	100	99.9	99.9	99.4	99.9	99.2	100	98.7
Saxophone	100	99.9	100	99.8	76.2	99	81.4	100
Vocal	100	98.8	97.8	100	99.4	99.8	99.6	100

Table 3. Coupling strategy for the two implementation (16-16) and (24-24) as compared (in percentage %) to the floating point version. By use of block exponent method the accuracy of the 16-16 version is much improved compared to the figures in Table 2.

Accordingly, as shown in Figure 5, the coupling co-ordinates may be calculated using 16-bit values only. The pre-processing stage of the 32-bit numbers before truncation again serves to improve results appreciably. From Table 4, below, it is evident that both the 24-24 and the 16-32 versions have similar performance.

	Drums		Harp		Piano		Saxophone		Vocal	
	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ
16-32	0.04	0.23	0.08	0.31	0.12	0.36	0.23	0.44	0.05	0.27
24-24	0.04	0.22	0.09	0.33	0.12	0.37	0.04	0.23	0.04	0.28

Table 4. Mean ($\bar{e}$) and standard deviation (σ) of the error between the floating point - and 16-16 (with block exponent) and 24-24 version. The figures are almost the same for both

implementations.

Rematrixing

The upper 16-bits of the 32-bit data resulting from the frequency transformation stage may be utilised to determine rematrixing for each band, in a manner similar to the coupling phase estimation. Within each rematrixing band, power measurements are made for the left channel (L), right channel (R); and the channel resulting from the sum (L+R) and difference (L-R).

If the maximum power is found in the L+R or L-R signal, then the rematrix flag is set and for that band, and L+R and L-R are encoded instead of L and R. For the encoding process full 32-bit data is used to provide maximum accuracy.

If the maximum power is in L or R, the rematrixing flag is not set for that band and the 32-bit data moves directly to the encoding process. Table 5 below compares the 16-bit (as just described) to the floating point version. The high figures indicate that for computing the rematrixing flag, the above described block exponent method is not necessary.

	Band 0		*Band 1*		*Band 2*		*Band 3*	
	16-16	*24-24*	*16-16*	*24-24*	*16-16*	*24-24*	*16-16*	*24-24*
Drums	*100*	*100*	*99.6*	*100*	*99*	*99.6*	*99.6*	*100*
Harp	*95.3*	*97.6*	*95.3*	*99.4*	*97*	*98.8*	*94.7*	*97.6*
Piano	*99.3*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
Saxophone	*96.3*	*98.9*	*97*	*99.2*	*99.2*	*99*	*99.6*	*100*
Vocal	*99.6*	*100*	*100*	*100*	*99.3*	*100*	*100*	*100*

Table 5. Comparison (in percentage %) of the rematrixing flag for the floating point - and 16-16 (without block exponent) and 24-24 version. The high figures (94% - 100%) for the 16-16 indicate that block exponent procedure is not very necessary.

Results

Figures 7, 8 and 9 are frequency response charts in terms of signal-to-noise ratio for the three discussed implementations, namely floating point, 24-24 bit and 16-32 bit calculations, respectively. This result is obtained by encoding-decoding 100 dB sinusoids at discrete frequency points, for the encoder version in question. The output from the decoder is compared with the original sinusoid to estimate the SNR. Note from the graph that the floating-point version gives average SNR of 85 dB (16-bit PCM has SNR of 96 dB). The SNR measurement does not take the masking and psychoacoustic effects in consideration, but nevertheless gives a number with which to compare different implementations. The frequency response shown in Figure 8 is of the 24-24 AC-3 encoder, which implies that for all processing single precision arithmetic with register length of 24-bit was assumed. On the other hand, the frequency response shown in Figure 9 is of the 16-32 AC-3 encoder, which in this context implies: 16-16 for transient detection, 16-32 for windowing, 32-16 for Frequency Transformation, 16-16 for coupling (determining phase and coupling co-ordinate), 32-32 for coupling channel generation, 16-16 for calculation of rematrixing flag and 32-32 for the rematrixing process.

Claims:

1. A method for coding digital audio data with a transform encoding process implemented on a fixed point digital signal processor having multiple levels of computation precision, wherein the transform encoding process includes a plurality of computation stages involving arithmetic operations in transforming the digital audio data into coded audio data, and wherein different ones of the computation stages utilise different preselected levels of computational precision, characterised in that:

the transform encoding process is in accordance with AC-3 Digital Audio Compression Standard.

2. A method as claimed in claim 1, wherein the digital signal processor comprises a 16-bit digital signal processor which is capable of single (16-bit) precision computations and double (32-bit) computations.

3. A method as claimed in claim 1 or 2, wherein the plurality of computation stages includes transient detection, windowing, frequency transformation, coupling strategy determination and coupling channel computation, and rematrixing determination and computation.

4. A method as claimed in claim 1 or 2, wherein the transform encoding process includes a transient detection process for detecting transients in the audio data, and wherein the transient detection process is carried out with single precision computations.

5. A method as claimed in claim 1 or 2, wherein the transform encoding process includes a windowing function which is carried out with single precision audio data and double precision coefficients.

6. A method as claimed in claim 1 or 2, wherein the transform encoding process includes a windowing function which is carried out with double precision audio data and single precision coefficients.

7. A method as claimed in claim 1 or 2, wherein the transform encoding process includes a frequency transformation process which is performed with double precision data and single precision coefficients.

8. A method as claimed in claim 1 or 2, wherein the transform encoding process includes determination of a coupling strategy and/or a phase strategy, and wherein the determination is performed with single precision data.

9. A method as claimed in claim 8, wherein the determination of coupling and/or phase strategy includes pre-processing by use of a block exponent method, wherein double precision frequency coefficients are shifted to eliminate leading zeros and truncated to single precision.

10. A method as claimed in claim 8 or 9, wherein the transform encoding process includes the formation of a coupling channel which is performed with double precision data.

11. A method as claimed in claim 1 or 2, wherein the transform encoding process includes a rematrixing determination which is performed with single precision data, and a rematrix coding process which is performed with double precision data.

12. A digital audio transform encoder for coding digital audio data into compressed audio data, comprising a fixed point digital signal processor having multiple levels of computation precision, and transform encoding process code stored in firmware or software for controlling the digital signal processor, wherein the transform encoding process code includes a plurality of computation blocks involving arithmetic operations in transforming the digital audio data into compressed audio data, and wherein different ones of the computation blocks are performed by the digital signal processor using different preselected levels of computational precision, characterised in that:

the transform encoding process code is in accordance with AC-3 Digital Audio Compression Standard.

13. An audio transform encoder as claimed in claim 12, wherein the digital signal processor comprises a 16-bit digital signal processor which is capable of single (16-bit) precision computations and double (32-bit) computations.

14. An audio transform encoder as claimed in claim 12 or 13, wherein the plurality of computation blocks include transient detection, windowing, frequency transformation, coupling strategy determination and coupling channel computation, and rematrixing determination and computation.

15. An audio transform encoder as claimed in claim 12 or 13, wherein the transform encoding process code includes a transient detection block for detecting transients in the audio data, and wherein the transient detection block utilises single precision computations.

16. An audio transform encoder as claimed in claim 12 or 13, wherein the transform encoding process code include a windowing block which utilises single precision audio data and double precision coefficients.

17. An audio transform encoder as claimed in claim 12 or 13, wherein the transform encoding process code includes a windowing block which utilises double precision audio data and single precision coefficients.

18. An audio transform encoder as claimed in claim 12 or 13, wherein the transform encoding process code includes a frequency transformation block which utilises double precision data and single precision coefficients.

19. An audio transform encoder as claimed in claim 12 or 13, wherein the transform encoding process code includes a block for determination of a coupling strategy and/or a phase strategy, and wherein the determination utilises single precision data.

20. An audio transform encoder as claimed in claim 19, wherein the block for determination of coupling and/or phase strategy utilises pre-processing by use of a block exponent method, wherein double precision frequency coefficients are shifted to eliminate leading zeros and truncated to single precision.

AMENDED SHEET

21. An audio transform encoder as claimed in claim 19 or 20, wherein the transform encoding process code includes a block for the formation of a coupling channel which utilises double precision data.

22. An audio transform encoder as claimed in claim 12 or 13, wherein the transform encoding process code includes a rematrixing determination block which utilises single precision data, and a rematrix coding block which utilises double precision data.

AMENDED SHEET

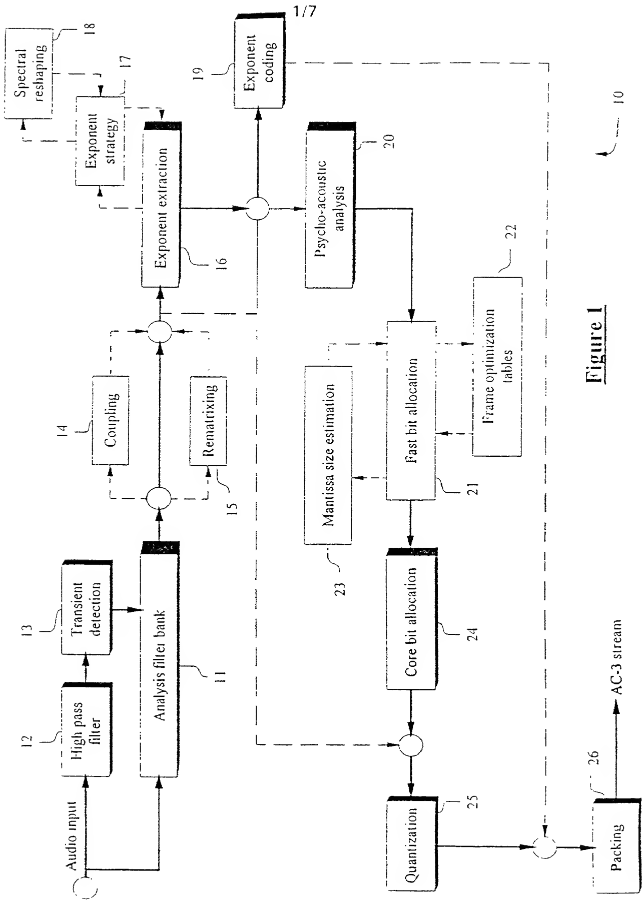

Audio input
High pass filter
12
Transient detection
13
Analysis filter bank
11
Coupling
14
Rematrixing
15
Exponent extraction
16
Exponent strategy
17
Spectral reshaping
18
Exponent coding
19
Psycho-acoustic analysis
20
Fast bit allocation
21
Frame optimization tables
22
Mantissa size estimation
23
Core bit allocation
24
Quantization
25
Packing
26
AC-3 stream
10

Figure 1

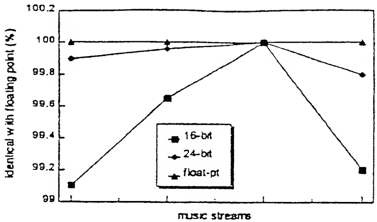


Figure 2

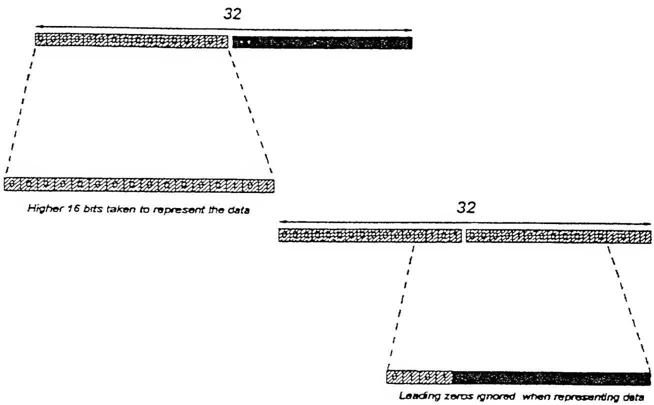


Figure 6

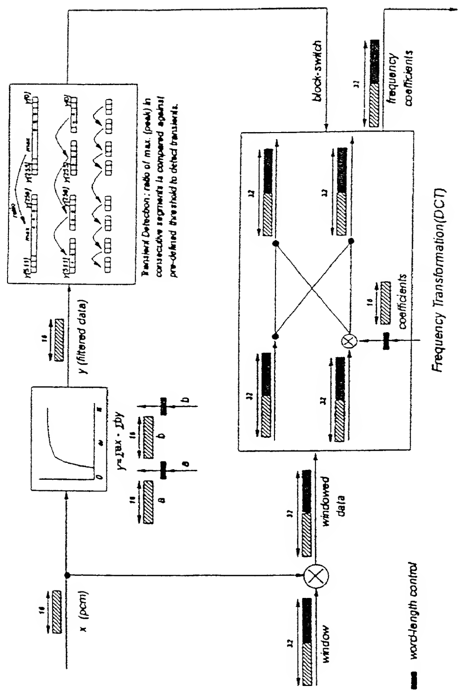
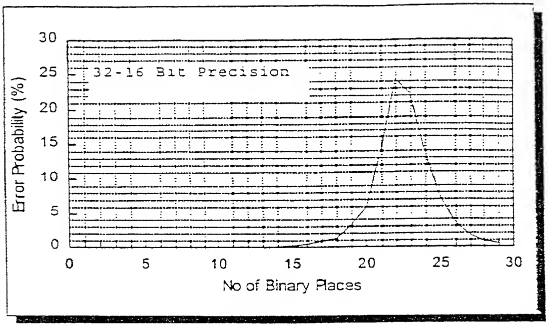

x (pcm)
16
window
32
windowed data
word-length control
y = Σax - Σby
a
b
y (filtered data)
Transient Detection: ratio of max. (peak) in consecutive segments is compared against pre-defined threshold to detect transients.
block-switch
frequency coefficients
coefficients
Frequency Transformation(DCT)

Figure 3


32-16 Bit Precision
Error Probability (%)
No of Binary Places
0
5
10
15
20
25
30

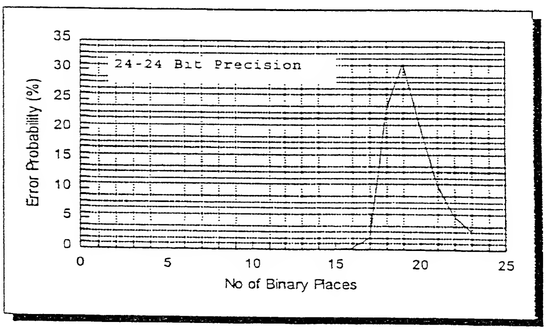

24-24 Bit Precision
Error Probability (%)
No of Binary Places
0
5
10
15
20
25
30
35

Figure 4

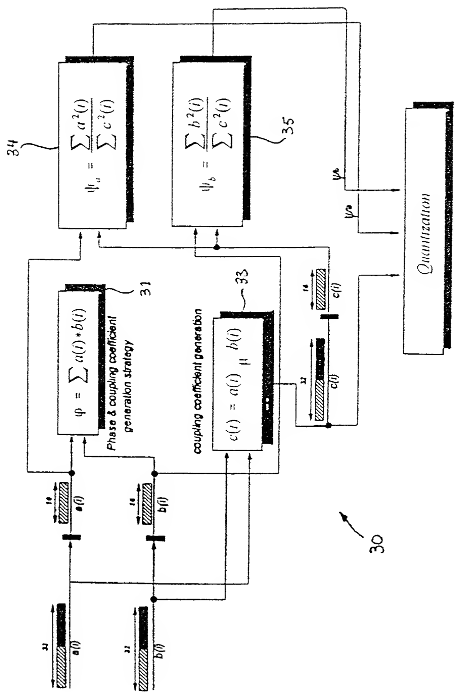

Phase & coupling coefficient generation strategy
coupling coefficient generation
Quantization
31
33
34
35
30

Figure 5

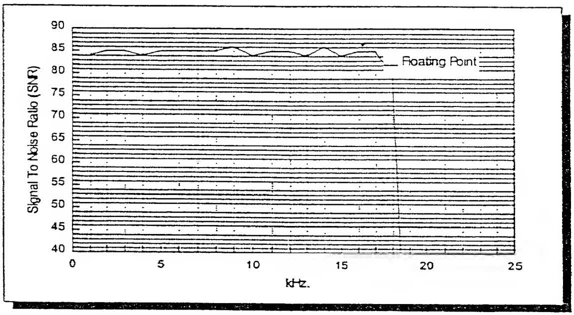

90
85
80
75
70
65
60
55
50
45
40
0
5
10
15
20
25
kHz.
Signal To Noise Ratio (SNR)
Floating Point

Figure 7

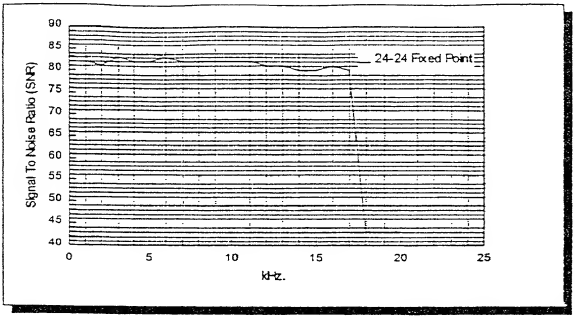


Figure 8

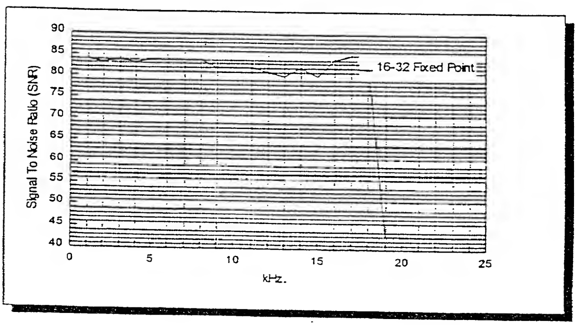


Figure 9

DECLARATION AND POWER OF ATTORNEY

As the below-named inventors, we declare that:

Our residences, post office addresses, and citizenships are as stated below under our names.

We believe we are the original, first, and joint inventors of the invention entitled "MULTI-PRECISION TECHNIQUE FOR DIGITAL AUDIO ENCODER," which is described and claimed in the specification and claims of International Patent Application No. PCT/SG98/00084, which was filed on 26 October 1998 and for which a patent is sought.

We have reviewed and understand the contents of the foregoing specification, including the claims, as amended by any amendment specifically referred to herein (if any).

We acknowledge our duty to disclose information of which we are aware which is material to the patentability and examination of this application in accordance with 37 C.F.R. § 1.56(a).

We hereby claim foreign priority benefits under 35 U.S.C. § 119 of the foreign patent application listed below:

PRIOR FOREIGN/PCT APPLICATION(S) AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. 119:			
COUNTRY	APPLICATION NUMBER	DATE OF FILING	PRIORITY CLAIMED UNDER 35 USC 119
PCT	PCT/SG98/00084	26 October 1998	Yes

We hereby appoint DAVID V. CARLSON, Registration No. 31,153; MICHAEL J. DONOHUE, Reg. No. 35,859; ROBERT IANNUCCI, Reg. No. 33,514; E. RUSSELL TARLETON, Reg. No. 31,800; ERIC J. GASH, Reg. No. 46,274; KEVIN S. COSTANZA, Registration No. 37,801; SUSAN D. BETCHER, Reg. No. 43,498; BRIAN L. JOHNSON, Registration No. 40,033; GEORGE C. RONDEAU, JR., Reg. No. 28,893; BRIAN G. BODINE, Reg. No. 40,520; CHARLES J. RUPNICK, Reg. No. 43,068; TIMOTHY L. BOLLER, Reg. No. 47,435; and FRANK ABRAMONTE, Reg. No. 38,066; comprising the firm of Seed Intellectual Property Law Group PLLC, 701 Fifth Avenue, Suite 6300, Seattle, Washington 98104-7092; and THEODORE E. GALANTHAY, Registration No. 24,122; LISA K. JORGENSON, Registration No. 34,845; ROBERT D. McCUTCHEON, Registration No. 38,717; and MARIO DONATO, Reg. No. 37,816; as our attorneys to prosecute this application and to transact all business in the U.S. Patent and Trademark Office in

17

connection therewith. Please direct all telephone calls to Eric J. Gash at (206) 622-4900 and telecopies to (206) 682-6031.

We further declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that the making of willfully false statements and the like is punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and may jeopardize the validity of any patent issuing from this patent application.

1-00

Javed Absar

Mohammed Javed Absar

Date 9th May 2001

Residence	:	City of Singapore
		~~Country of Singapore~~ SGX
Citizenship	:	India
P.O. Address	:	Block 411, #09-1006 Hougang Avenue 10
		Singapore 530411

2-00

Sapna George

Sapna George

Date 9th May 2001

Residence	:	City of Singapore
		~~Country of Singapore~~ SGX
Citizenship	:	India
P.O. Address	:	Block 315, #06-220 Serangoon Avenue 2
		Singapore 550315

3ω

Anotonio Mario Alvarez-Tinoco

Date 9/5/01

Residence : City of Singapore
Country of Singapore SGX

Citizenship : United Kingdom

P.O. Address : 32 Toh Tuck Road #03-01
Singapore 596710

F:\Foreign-National\US PCT declaration power attorney joint inventor ST.doc